10-725/36-725: Convex Optimization **Fall 2016**

Lecture 22: November 16

Lecturer: Javier Pena *Scribes: Shuo Zhao,Sanghamitra Dutta,Yohan Jo*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

22.1 A Recap of ADMM

Consider a problem of the form

$$\min_{z,x} f(x) + g(z) \text{ subject to} Ax + Bz = c \tag{22.1}$$

The augmented Lagrangian (for some $\rho > 0$) takes the form:-

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2 \tag{22.2}$$

The ADMM steps for $k = 1, 2, \ldots,$ is given by,

$$x^{(k)} = \arg\min_x L_\rho(x, z^{(k-1)}, u^{(k-1)}) \tag{22.3}$$

$$z^{(k)} = \arg\min_z L_\rho(x^{(k)}, z, u^{(k-1)}) \tag{22.4}$$

$$u^{(k)} = \arg\min_u L_\rho(x^{(k)}, z^{(k)}, u) \tag{22.5}$$

22.2 ADMM in scaled form

We can replace the dual variable u by a scaled variable $w = \frac{u}{\rho}$. In this parametrization, the ADMM steps are

$$x^{(k)} = \arg\min_x f(x) + \frac{\rho}{2}||Ax + Bz^{k-1} - c + w^{k-1}||_2^2 \tag{22.6}$$

$$z^{(k)} = \arg\min_z g(z) + \frac{\rho}{2}||Ax^{(k)} + Bz - c + w^{k-1}||_2^2 \tag{22.7}$$

$$w^{(k)} = w^{k-1} + Ax^{(k)} + Bz^{(k)} - c+ \tag{22.8}$$

22.3 Projected Gradient Descent

Consider the constrained optimization problem,

$$\min_x f(x) \text{ subject to } x \in C \tag{22.9}$$

where $f(x)$ is convex and smooth, and C is convex. Let us recall projected gradient descent.

Choose an initial x^0, and for, $k = 1, 2, 3, \ldots,$

$$x^{(k)} = P_C\left(x^{(k-1)} - t_k \nabla f(x^{(k-1)})\right) \tag{22.10}$$

Projected Gradient Descent is essentially a particular case of proximal gradient descent. It is motivated by a local quadratic expansion of f as follows:

$$x^{(k)} = P_C\left(\arg\min_y \nabla f(x^{(k-1)})^T(y - x^{(k-1)}) + \frac{1}{2t}||y - x^{(k-1)}||_2^2\right) \tag{22.11}$$

22.4 Conditional Gradient (Frank-Wolfe) Method

Rather than minimizing a quadratic approximation, we are going to do something simpler. First look at the point in C, that minimizes dot product with the gradient of $f(x)$. Essentially, we make a trade-off. Instead of a projection, we minimize a linear function over C, which is simpler and more efficient in many cases.

In principle, we could do a line search. We could take a convex combination from current point to minimizer.

Let us now state the method formally.

Choose an initial x^0, and for, $k = 1, 2, 3, \ldots,$

$$s^{(k-1)} \in \arg\min_{s \in C} \nabla f(x^{(k-1)})^T s \tag{22.12}$$

$$x^{(k)} \in (1 - \gamma_k)x^{(k-1)} + \gamma_k s^{(k-1)} \tag{22.13}$$

Note that there is no projection; update is solved directly over the constraint set C. The default choice for step sizes is $\gamma_k = \frac{2}{(k+1)}$ for $k = 1, 2, \ldots$. For any choice $0 \leq \gamma_k \leq 1$, we see that $x^{(k)} \in C$ by convexity. Can also think of the update as,

$$x^{(k)} \in x^{(k-1)} + \gamma_k(s^{(k-1)} - x^{(k-1)}) \tag{22.14}$$

i.e., we are moving less and less in the direction of the linearization minimizer as the algorithm proceeds.

In many cases, sub gradient is easier than projection on a ball L1 Special case of co-ordinate descent.

Interesting Remark: Frank and Wolfe were post-docs working with Tucker. They proposed this algorithm first for quadratic functions. It was published in 56, and later there was a follow-up paper. After that, for many years, there were no more papers. In the last 6-7 years, again renewal of interest, particularly due to the insights of Jaggi.

22.4.1 Norm Constraints

What happens when $C = \{x | ||x|| \leq t\}$ for a norm $||.||$?

$$s^{(k-1)} \in \arg\min_{s \in C} \nabla f(x^{(k-1)})^T s \tag{22.15}$$

$$= -t. \left(\arg\max_{||s|| \leq 1} \nabla f(x^{(k-1)})^T s \right) \tag{22.16}$$

$$= -t. \partial ||\nabla f(x^{(k-1)})||_* \tag{22.17}$$

where $||.||_*$ is the corresponding dual norm. In other words, if we know how to compute sub-gradients of the dual norm, then we can easily perform Frank-Wolfe steps

A key to Frank-Wolfe: this can often be simpler or cheaper than projection onto $C = \{x | ||x|| \leq t\}$. This is also often simpler or cheaper than the prox operator for $||.||$.

22.4.2 Examples

22.4.2.1 l_1 Regularization

For the l_1 regularized problem, we have

$$\min_x f(x) \text{subject to } ||x||_1 \leq t \tag{22.18}$$

We have $s^{(k-1)} \in -t. \partial ||\nabla f(x^{(k-1)})||_*$. Frank-Wolfe update is thus,

$$i_{(k-1)} = \arg\max_{i=1,2,\ldots,p} |\nabla_i f(x^{(k-1)})| \tag{22.19}$$

$$x_{(k)} = (1 - \gamma_k) x^{(k-1)} - \gamma_k t. sign(\nabla_{i^{(k-1)}} f(x^{(k-1)})). e_{i^{(k-1)}} \tag{22.20}$$

This is a kind of coordinate descent. Note that, this is a lot simpler than projection onto the "l_1 ball", though both require $O(n)$ operations.

22.4.2.2 l_p Regularization

For the l_p regularized problem, we have

$$\min_x f(x) \text{subject to } ||x||_p \leq t \tag{22.21}$$

for $1 \leq p \leq \infty$, we have $s^{(k-1)} \in -t. \partial ||\nabla f(x^{(k-1)})||_q$ where q is the dual of p, *i.e.*, $\frac{1}{p} + \frac{1}{q} = 1$.

We claim that we can choose,

$$s_i^{(k-1)} = -\alpha sign(\nabla_{i^{(k-1)}} f(x^{(k-1)})) |\nabla_{i^{(k-1)}} f(x^{(k-1)})|^{\frac{p}{q}} \tag{22.22}$$

where α is a constant such that $|s^{(k-1)}|_q = t$, and then the Frank-Wolfe updates follow as usual.

Note that, this is a lot simpler than projection onto the "l_p" ball, for any general p. Aside from special cases ($p = 1, 2, \ldots, \infty$), these projections cannot be directly computed and must be treated as an optimization.

22.4.2.3 Trace Norm Regularization

For the trace-regularized problem,

$$\min_X f(X) \text{subject to } ||X||_{tr} \leq t \tag{22.23}$$

we have, $S^{(k-1)} \in -t.\partial||\nabla f(X^{(k-1)})||_{op}$.

We claim that, we can choose

$$s_i^{(k-1)} = -t.uv^T \tag{22.24}$$

where u, v are the leading left and right singular vectors of $\nabla f(X^{(k-1)})$, and then the Frank-Wolfe updates follow as usual.

Note that, this is a lot simpler and more efficient than projection onto the trace norm ball, which requires a singular value decomposition.

22.5 Constrained and Lagrange forms

Recall the solutions of the constrained problem

$$\min_x f(x) \text{ subject to} ||X|| \leq t \tag{22.25}$$

are equivalent to those of the Lagrange problem.

$$\min_x f(x) + \lambda||X|| \tag{22.26}$$

Now we compare the Frank-Wolfe updates under $||\cdot||$ to proximal operator of $||\cdot||$.

For ℓ_2 norm, Frank-Wolfe update scans for maximum of gradient while proximal operator soft-thresholds the gradient step. Both methods use O(n) flops.

For ℓ_p norm, Frank-Wolfe update raises each entry of gradient to power and sums in O(n) flops while proximal operator is not generally directly computable.

For Trace norm, Frank-Wolfe update computes top left and right singular vectors of gradient while proximal operator soft-thresholds the gradient step with SVD.

There are many cases Frank-Wolfe updates are very efficient such as special polyhedra or cone constraints, sum-of-norms (group-based) regularization, atomic norms. See Jaggi (2011) for more details.

As an example, we compare the performance of projected and conditional gradient for constrained lasso problem with n=100 and p=500. As shown in the figure, conditional gradient methods match the convergence rate with first order methods. However, they can be slower to converge to high accuracy in practice. In the future, conditional gradient methods may beat projected gradient methods, which Prof. Javier with his collaborators have been working on.

22.6 Duality Gap

Naturally, Frank-Wolfe iterations admit a duality gap which is indeed a suboptimality gap:

$$\max_{s \in C} \nabla f(x^{(k-1)})^T (x^{(k-1)} - s) \tag{22.27}$$

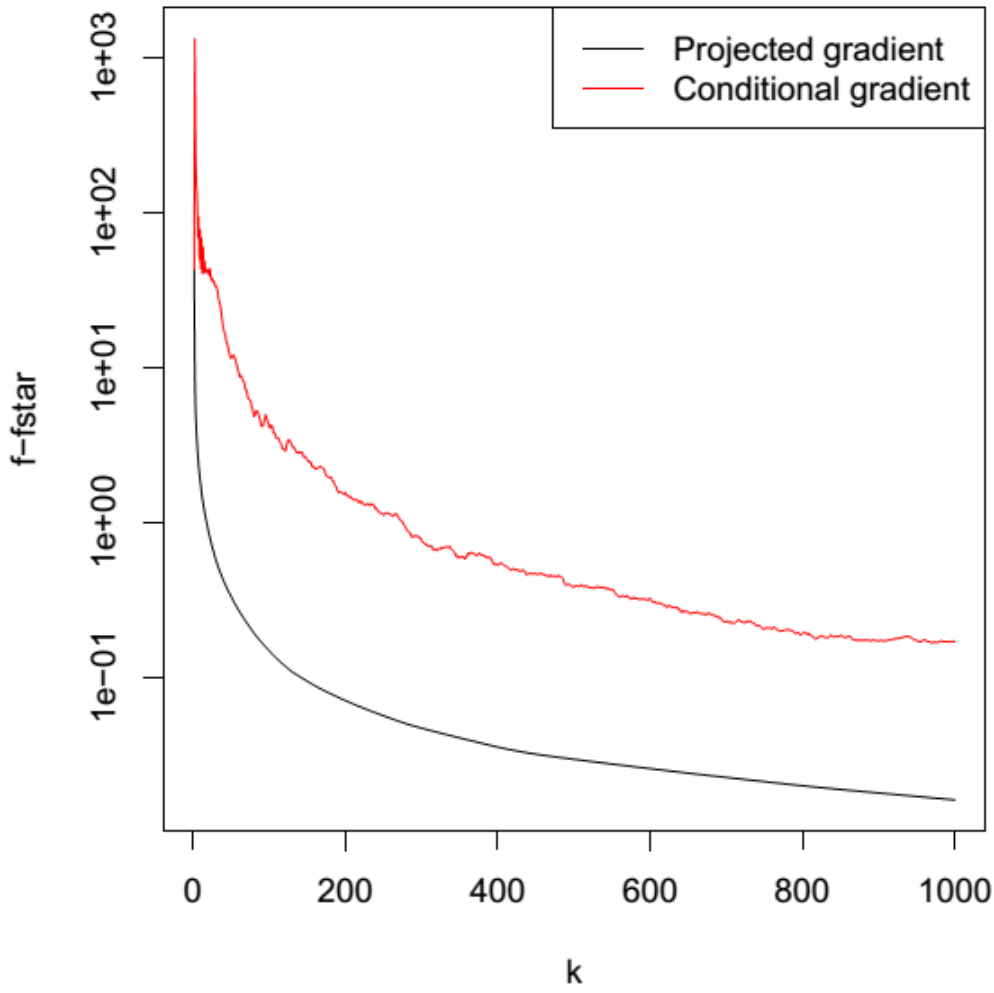

Figure 22.1: Projected and Conditional gradient for constrained lasso problem

This is an upper bound on $f(x^{(k-1)}) - f^\star$. Now we prove this. Firstly, by the first-order condition for f being convex:

$$f(s) \geq f(x^{(k-1)}) + \nabla f(x^{(k-1)})^T (s - x^{(k-1)}) \tag{22.28}$$

Then we minimize on both sides over all $s \in C$:

$$f^\star \geq f(x^{(k-1)}) + \min_{s \in C} \nabla f(x^{(k-1)})^T (s - x^{(k-1)}) \tag{22.29}$$

Finally, we rearrange to show the duality gap is an upper bound.

$$\max_{s \in C} \nabla f(x^{(k-1)})^T (x^{(k-1)} - s) = f(x^{(k-1)})^T (x^{(k-1)} - s^{(k-1)}) \tag{22.30}$$

This quantity directly comes from the Frank-Wolfe update.

We can rewrite the original problem as

$$\min_x f(x) + I_C(x) \tag{22.31}$$

where I_C is the indicator function of C.

The dual problem is

$$\min_u -f^*(u) - I_C^*(-u) \tag{22.32}$$

where I_C^* is the support function of C.

Therefore, the duality gap at x, u is

$$f(x) + f^*(u) + I_C^*(-u) \geq x^T u + I_C^*(-u) \tag{22.33}$$

Evaluated at $x = x^{(k-1)}$, $u = \nabla f(x^{(k-1)})$, this will give the claimed gap. That's why we call it "duality gap".

22.7 Convergence analysis

In order to study the convergence property of Frank-Wolfe methods, we first need to define the curvature constant of f over C following Jaggi (2011) :

$$M = \max_{x,s,y \in C; y=(1-\gamma)x+\gamma s} 2(f(y) - f(x) - \nabla f(x)^T(y-x))/\gamma^2 \tag{22.34}$$

where γ is between 0 and 1. From the above definition, it is easy to see M actually measures how far you are away from linear approximation. $M = 0$ indicates f is linear.$f(y) - f(x) - \nabla f(x)^T(y-x)$ is called the Bregman divergence defined by f.

Theorem 22.1 *Conditional gradient method using fixed step sizes $\gamma_k = 2/(k+1), k = 1, 2, 3...$ satisfies:*

$$f(x^{(k-1)}) - f^\star \leq 2M/(k+2) \tag{22.35}$$

Number of iterations needed to have $f(x^{(k-1)}) - f^\star \leq \epsilon$ is $O(1/\epsilon)$.

Proof of the Theorem: We are going to prove this theorem by induction. Before we jump into the proof, we need to introduce a basic inequality.

$$f(x^{(k)}) \leq f(x^{(k-1)}) - \gamma_k g(x^{(k-1)}) + \gamma_k^2 M/2 \tag{22.36}$$

where $g(x) = \max_{s \in C} \nabla f(x)^T(x-s))$ is the duality gap mentioned in previous slides.

To prove this basic inequality, we write $x^+ = x$, $x = x^{(k-1)}$, $s = s^{(k-1)}$, $\gamma = \gamma_k$. Then

$$f(x^+) = f(x + \gamma(s-x)) \tag{22.37}$$
$$\leq f(x) + \gamma \nabla f(x)^T(s-x) + \gamma^2 M/2 \tag{22.38}$$
$$= f(x) - \gamma g(x) + \gamma^2 M/2 \tag{22.39}$$

The second line uses the definition of M and the last line uses the definition of g.

Now, with the basic inequality, we are using induction to prove the convergence rate theorem.

For the $k = 1$ case, it is easy to check the theorem holds.

For arbitrary $k > 1$, we assume $f(x^{(k-1)}) - f^\star \leq 2M/(k+1)$ holds.

Now apply the basic inequality theorem:

$$f(x^{(k)}) \leq f(x^{(k-1)}) - \gamma_k g(x^{(k-1)}) + \gamma_k^2 M/2 \tag{22.40}$$

Recall the duality gap $g(x)$, we have $g(x^{(k-1)}) \leq f(x^{(k-1)}) - f^\star$ and $\gamma_k = 2/(k+1)$.

Thus, take them back into the equality:

$$f(x^{(k)}) \leq f(x^{(k-1)}) - 2(f(x^{(k-1)}) - f^\star)/(k+1) + 4M/2(k+1)^2 \tag{22.41}$$

Hence,

$$f(x^{(k)}) - f^\star \leq (1 - 2/(k+1))(f(x^{(k-1)}) - f^\star) + 2M/(k+1)^2 \tag{22.42}$$

Using induction:

$$f(x^{(k)}) - f^\star \leq (k - 1/k + 1) \times 2M/(k+1) + 2M/(k+1)^2 \leq 2M/(k+2) \tag{22.43}$$

Now, we have finished the proof by induction.

This proved convergence rate matches the known rate for projected gradient descent when ∇f is Lipschitz. Now we compare the assumptions. Actually, if ∇f is Lipschitz with constant L then $M \leq diam^2(C) \cdot L$, where $diam^2(C) = \max_{x,s \in C} ||x - s||_2$

To see this, recall if ∇f is Lipschitz with constant L, we have

$$f(y) - f(x) - \nabla f(x)^T (y - x) \leq L/2||y - x||_2^2 \tag{22.44}$$

Maximizing over all $y = (1-\gamma)x + \gamma x$, and ,multiplying by $2/\gamma^2$,

$$M \leq \max_{x,s,y \in C; y = (1-\gamma)x + \gamma s} 2/\gamma^2 \times L/2||y - x||_2^2 = \max_{x,s \in C} L||x - s||_2^2 \tag{22.45}$$

and the bound follows. Essentially, assuming a bounded curvature is no stronger than what we assumed for proximal gradient.

22.8 Affine invariance

Recall that

- Gradient Descent: $x^+ = x - t\nabla f(x)$
- Pure Newton's Method: $x^+ = x - \nabla^2 f(x)^{-1} \nabla f(x)$.

Gradient descent is not affine invariant, i.e., scaling the coordinates may make gradient descent perform better. In contrast, Newton's method is affine invariant, i.e., the algorithm behaves the same given any affine transformation of the variables. Conditional gradient method has a similar spirit to gradient descent, but it is affine invariant.

Given nonsingular $A : \mathbb{R}^n \to \mathbb{R}^n$, $x = Ax'$, and $h(x') = f(Ax')$, affine invarnace of conditional gradient method means

$$\min f(x) \text{ subject to } x \in C \quad \Leftrightarrow \quad \min h(x') \text{ subject to } x' \in A^{-1}C.$$

Frank-Wolfe on $h(x')$ proceeds as

$$s' = \operatorname*{argmin}_{z \in A^{-1}C} \nabla h(x')^T z$$
$$(x')^+ = (1-\gamma)x' + \gamma s'$$

Even convergence analysis is affine invariant. The curvature constant M of h is

$$M = \max_{\substack{x',s',y' \in A^{-1}C \\ y'=(1-\gamma)x'+\gamma s'}} \frac{2}{\gamma^2} \left(h(y') - h(x') - \nabla h(x')^T (y' - x') \right)$$

matching that of f, because $\nabla h(x')^T (y' - x') = \nabla f(x)^T (y - x)$.

However, affine invariance does not come intuitive in the bound of M

$$M \leq \max_{x,s \in C} L||x - s||_2^2,$$

given that the diameter of C on the righthand side is not affine invariant. This is Worth pondering!

22.9 Inexact updates

Suppose we choose $s^{(k-1)}$ so that

$$\nabla f(x^{(k-1)})^T s^{(k-1)} \leq \min_{s \in C} \nabla f(x^{(k-1)})^T s + \frac{M\gamma_k}{2} \delta,$$

where $\delta \geq 0$ is an inaccuracy parameter. Then we attain the same convergence rate.

Theorem 22.2 *Conditional gradient method using fixed step sizees $\gamma_k = 2/(k+1), k = 1, 2, 3, ...$, and inaccuracy parameter $\delta \geq 0$, satisfies*

$$f(x^{(k)} - f^\star \leq \frac{2M}{k+2}(1 + \delta).$$

22.10 Some variants

Some variants of the conditional graident method:

- Line search: In certain cases, we can do line search instead of fixing $\gamma_k = 2/(k+1), k = 1, 2, 3, ...$, use exact line search for the step sizes

  $$\gamma_k = \operatorname*{argmin}_{\gamma \in [0,1]} f\left(x^{(k-1)} + \gamma(s^{(k-1)} - x^{(k-1)}) \right)$$

  at each $k = 1, 2, 3, ...$ Or, we could use backtracking.

  For example, if f is a quadratic function, line search is easy and leads to faster convergence.

- Fully corrective: We update x directly according to

  $$x^{(k)} = \operatorname*{argmin}_y f(y) \text{ subject to } y \in \text{conv}\{x^{(0)}, s^{(0)}, ..., s^{(k-1)}\}$$

  The fully corrective way can make better progress, but is much more expensive.

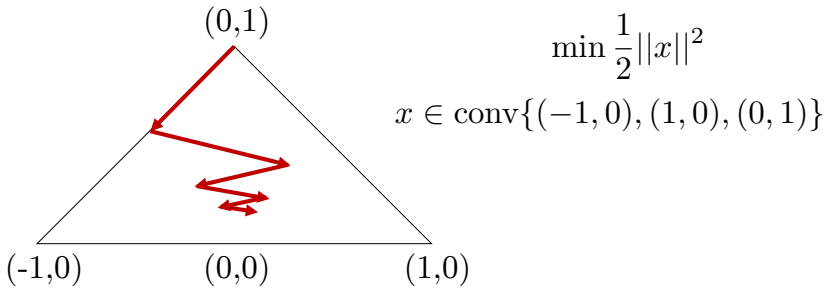

Figure 22.2: Away step motivation

22.11 Away steps

For motivation, let's look at the minimization problem in Figure 22.2, where the optimal solution is at (0,0). The conditional descent method is "jammed" because of the initial point (0,1). Conditional gradient descent with away steps moves not only toward a likely point, but also away from an unlikely point.

Suppose $C = \text{conv}(A)$ for a set of atoms A. Keep explicit description of $x \in C$ as a convex combination of elements in A

$$x = \sum_{a \in A} \lambda_a(x) a.$$

Conditional gradient with away steps:

1. Choose $x^{(0)} = a^{(0)} \in A$
2. For $k = 1, 2, 3, ...$
   - $s^{(k-1)} \in \text{argmin}_{a \in A} \nabla f(x^{(k-1)})^T a$
   - $a^{(k-1)} \in \text{argmax}_{\substack{a \in A \\ \lambda_a(x^{(k-1)}) > 0}} \nabla f(x^{(k-1)})^T a$
   - Choose $v = s^{(k-1)} - x^{(k-1)}$ (regular step) or $v = x^{(k-1)} - a^{(k-1)}$ (away step).
   - $x^{(k)} = x^{(k-1)} + \gamma_k v$

22.12 Linear convergence

Consider the unconstrained problem

$$\min_x f(x) \text{ subject to } x \in \mathbb{R}^n,$$

where f is μ-strongly convex and $\nabla f$is L-Lipschitz.

For $t_k = 1/L$, gradient descent iterates $x^{(k+1)} = x^{(k)} - t_k \nabla f(x^{(k)})$ satisfy

$$f(x^{(k)}) - f^\star \leq \left(1 - \frac{\mu}{L}\right)^k (f(x^{(0)}) - f^\star).$$

Consider the constrained problem

$$\min_x f(x) \text{ subject to } x \in \text{conv}(A) \subseteq \mathbb{R}^n,$$

where A is finite.

Theorem 22.3 *Assume f is μ-strongly convex, ∇f is L-Lipschitz, and $A \subseteq \mathbb{R}^n$ finite. For suitable γ_k, the iterates generated by the conditional gradient algorithm with away steps satisfy*

$$f(x^{(k)}) - f^\star \leq (1-r)^{k/2}(f(x^{(0)}) - f^\star) \text{ for } r = \frac{\mu}{L}\frac{\Phi(A)^2}{4\text{diam(A)}^2},$$

where

$$\Phi(A) = \min_{F \in \text{faces(conv(A))}} \text{dist(F, conv(A\F))}.$$

If the polytope is flat, Φ is small and the algorithm converges slowly.

22.13 Path following

Given the norm constrained problem

$$\min_x f(x) \text{ subject to } ||x|| \leq t,$$

the Frank-Wolfe algorithm can be used for path following, i.e., can produce a (approximate) solution path $\hat{x}(t), t \geq 0$. Beginning at $t_0 = 0$ and the only feasible point $x^\star(0) = 0$, we fix parameters $\epsilon, m > 0$, then repeat for $k = 1, 2, 3, ...$:

- Calculate
$$t_k = t_{k-1} + \frac{(1-1/m)\epsilon}{||\nabla f(\hat{x}(t_{k-1}))||_*}$$
and set $\hat{x}(t) = \hat{x}(t_{k-1})$ for all $t \in (t_{k-1}, t_k)$.
- Compute $\hat{x}(t_k)$ by running Frank-Wolfe at $t = t_k$, terminating when the duality gap is less than ϵ/m.

With this path following strategy, we are guaranteed that

$$f(\hat{x}(t)) - f(x^\star(t)) \leq \epsilon \text{ for all } t \text{ visitied},$$

i.e., we produce a (piecewise-constant) path with suboptimality gap uniformly bounded by ϵ, over all t. The reason is that duality gap

$$g_t(x) = \max_{||s|| \leq 1} \nabla f(x)^T(x - s) = \nabla f(x)^T x + t||\nabla f(x)||_*.$$

has a linear property. If we keep the duality gap small enough, e.g., t_{k-1}, then we can increase t_{k-1} to t_k and still maintain the duality gap less than ϵ for the same x.

References